Department of Computer Science and Engineering
University of Texas at Arlington

System Test Plan

Team: 4Loop

Project: Smart Fitness Trainer

Team Members:
Thanuja Fernando
Andrew Gallagher
Thomas Heinen
Mark Ragunton

Last Updated:07/22/2013 10:51pm

Contents

Document Revision History 4
List of Figures 5
List of Tables 6
1. Introduction 8
1.1 Document Overview 8
1.2 Product Overview 8
1.3 Product Scope 8
2. References 9
2.1 Overview 9
2.2 System Requirements Specification 9
2.3 Architecture Design Specification 13
2.4 Detailed Design Specification 18
3. Test Items 23
3.1 Overview 23
3.2 Relational Diagram 24
3.3 Hardware Tests 25
3.4 Unit Tests 25
3.5 Component Tests 28
3.6 Integration Tests 31
3.7 System Verification Tests 34
4. Risks 35
4.1 Overview 35
4.2 Risk Table 35
5. Features to Be Tested 36
5.1 Overview 36
5.2 Customer Requirements 36
5.3 Performance Requirements 41
5.4 Maintenance and Support Requirements 43
5.5 Other Requirements 43
6. Features Not to Be Tested 44
6.1 Overview 44

6.2 Packaging Requirements 44
6.3 Safety Requirements 45
6.4 Other Requirements 45
7. Overall Test Strategy 47
7.1 Overview 47
7.2 Configurations 47
7.3 Strategy 47
7.4 Metrics 47
7.5 Regression 48
8. Acceptance Criteria 49
8.1 Overview 49
8.2 Hardware Testing 49
8.3 Unit (Module) Testing 49
8.4 Component (Subsystem) Testing 49
8.5 Integration (Layer) Testing 50
8.6 System Verification Testing 50
9. Test Deliverables 51
9.1 Overview 51
9.2 Deliverables 51
10. Test Schedule 53
10.1 Overview 53
10.2 Test Schedule 53
11. Approvals 54

Document Revision History

Revision Number	Revision Date	Description	Rationale
0.1	07/04/2013	Initial Draft	Initial Draft
1.0	07/12/2013	Peer Review Draft	Peer Review Draft
2.0	07/22/2013	Revisions based on review and final updates	Baseline Draft

List of Figures

Figure #	*Title*	*Page #*
2-1	Architecture Layers	14
2-2	Subsystem Data Flow Diagram	16
2-3	Module Data Flow Diagram	18
3-1	Relational Diagram	24

List of Tables

Table #	*Title*	*Page #*
2-1	Customer Requirements	9
2-2	Packaging Requirements	11
2-3	Performance Requirements	11
2-4	Safety Requirements	12
2-5	Maintenance & Support Requirements	12
2-6	Other Requirements	13
2-7	Subsystem Overview	15
2-8	Subsystem Data Flows	16
2-9	Input Layer Data Flows	18
2-10	Workout Processing Layer Data Flows	19
2-11	Profile Processing Layer Data Flows	20
2-12	Database Management Layer Data Flows	20
2-13	Presentation Layer Data Flows	21
2-14	Module Requirement Mapping	21
3-1	Hardware Tests	25
3-2	Input Layer Unit Tests	25
3-3	Workout Processing Layer Unit Tests	25
3-4	Profile Processing Layer Unit Tests	26
3-5	Database Management Layer Unit Tests	28
3-6	Presentation Layer Unit Tests	28
3-7	Input Layer Component Tests	29
3-8	Workout Processing Layer Component Tests	29
3-9	Profile Processing Layer Component Tests	30
3-10	Database Management Layer Component Tests	31

3-11	Presentation Layer Component Tests	31
3-12	Input Layer Integration Tests	32
3-13	Workout Processing Layer Integration Tests	32
3-14	Profile Processing Layer Integration Tests	32
3-15	Database Management Layer Integration Tests	33
3-16	Presentation Layer Integration Tests	33
3-17	System Verification Tests	34
4-1	Risks	35
10-1	Test Schedule	53
11-1	Approvals	54

1. Introduction

1.1 Document Overview

The purpose of the System Test Plan (STP) document is to describe the testing procedures we will use to ensure that the Smart Fitness Trainer system meets its necessary requirements. By following the testing strategy in this document, we will affirm that our product is usable and has the required functionality present when it is delivered to our customer Jeremy Roden.

The Detailed Design Specification (DDS) contained a brief section on Quality Assurance with a broad overview of testing considerations and procedures. In this STP document, we will pick up where we left off in the DDS by continuing this discussion in much more detail. We will provide detailed testing procedures for each module, subsystem, and layer in the Smart Fitness Trainer system. A section of this document will also be dedicated to an overall test strategy including the actual metrics we will use to ensure that the critical requirements from the System Requirements Specification (SRS) document have been met. The testing procedures covered in this document will include hardware, unit/module, subsystem/component, layer/integration, and system verification testing.

1.2 Product Overview

The Smart Fitness Trainer shall allow individuals to engage in a home work-out and be able to track their progression. At various points, the system shall use image analysis to obtain several key body statistics of the user, such as height, weight, and body mass index through the use of an RGB-D camera. Before a user is able to engage in a provided workout program, an initial readiness assessment will be performed. This assessment will require the user to take a medical assessment as well as perform short exercise routines as part of the fitness assessment to determine the user's current fitness level. After a user has been assessed on their health and performance, the system will generate a customized workout program designed specifically for the user. When a user is engaging in a workout program, the camera will detect the player's movements to verify that the user is performing the correct movements. Variables such as number of repetitions, number of sets, user's heart rate, and calories burned will be stored in order to track the user's progress. Upon completion of a program, all data will be saved into the SMART Trainer for future workouts for the user.

1.3 Product Scope

The Smart Fitness Trainer system is composed of pieces of hardware that allow an individual to engage in workout-related activities at home through the use of RGB-D cameras to recognize and analyze user movement. The system will be controlled and executed from a component built of several PC components such as a CPU, graphics processor, RAM, etc. and will be attached to the RGB-D to process and store all video data. The system will process the data which in turn will be used to generate a custom workout for an individual. A microphone will be used to capture audio and convert to text instructions, and a wireless keyboard will also be provided for input.

2. References

2.1 Overview

This STP document references several other documents that have gone into the creation of the Smart Fitness Trainer system including the SRS, ADS, and DDS. To guarantee that our system has been fully tested and verified, all of these documents and their contents must be taken into consideration during the development of the test plan. The following sections contain a brief overview of these other documents.

2.2 System Requirements Specification

In our SRS document, we defined all of the requirements that are necessary for our system. Thirty nine requirements were defined in this document, and broken down into categories (Customer, Packaging, Performance, Safety, Maintenance, and Other). We will give a brief description of each requirement and its assigned priority. For more information, see the System Requirements Specification.

2.2.1 Customer Requirements

This section describes the requirements which were most important to our project sponsor. Many of these requirements are critical for the core functionality of the Smart Fitness Trainer, and the project would be considered incomplete if they were not implemented.

Table 2-1 – Customer Requirements

SRS #	Requirement	Description	Priority
3.1	RGB-D Camera Tracking	The Smart Fitness trainer shall use a RGB-D camera to track the movement of the user.	1 - Critical
3.2	Movement Analysis	The Smart Fitness Trainer shall analyze the user's movements to determine if the user is performing an exercise correctly.	1 - Critical
3.3	Workout Progression Mode	The Smart Fitness Trainer shall allow the user to start Progression mode which will take the user through a medical check, fitness assessment, and allow the user to choose one of 3 fitness goals before generating a personalized workout.	1 - Critical
3.4	Medical Check	The Smart Fitness Trainer shall perform a medical check which will have the user answer commonly asked questions by trainers before starting a new workout.	1 - Critical
3.5	Fitness Assessment	The Smart Fitness Trainer shall have the user perform several basic exercises and measure the amount of correctly performed reps to provide a baseline of the user's prior workout experience.	1 - Critical
3.6	Workout Goals	The Smart Fitness Trainer shall allow the user to choose from one of three major workout	1 - Critical

		goals: Strength, Tone, and Cardio.	
3.7	Workout Statistics	The Smart Fitness Trainer shall track and display heart beat and calories burned (during the current workout), total performed reps, and correctly performed reps.	1 - Critical
3.8	Personalized Workout Plan	The Smart Fitness Trainer shall generate a personalized workout plan for the user based on the fitness assessment and workout goal.	1 - Critical
3.9	Workout Instructions	The Smart Fitness Trainer shall provide both audio and visual instructions to the user on how to perform a drill, and what they should be doing at certain times during their workout (ex. rest period, start, stop, etc.).	1 - Critical
3.10	Lifetime Workout Statistics	The Smart Fitness Trainer shall record the user's height, weight, BMI, total calories burned, minutes worked out, and other information about the user, which can be displayed in graph form.	2 - High
3.11	Quick Workout	The Smart Fitness Trainer shall allow the user to substitute any workout with a "quick workout," allowing them to select their own drills from the database for the workout.	4 – Low
3.12	User Interface	The Smart Fitness Trainer shall have an appealing user interface which will be intuitive for the user.	1 - Critical
3.13	User Profile	The Smart Fitness Trainer shall allow the user to create, edit or delete a user profile, which will be used to log in to progression mode.	2 – High
3.14	User Pictures	The Smart Fitness Trainer shall allow the user to take pictures of themselves at different points in the workout plan to assess how the workout has impacted their appearance.	3 – Moderate
3.15	Voice Commands	The Smart Fitness Trainer shall accept voice commands for menu interactions and other inputs during workouts.	2 – High
3.16	Personal Best Recognition	The Smart Fitness Trainer shall be able to identify when a user has performed an exercise better than any previous attempts, and display a notification of a "personal best."	2 - High

2.2.2 Packaging Requirements

This section defines the packaging requirements for the Smart Fitness Trainer system which includes the hardware and software components of the system, and any additional things which will be included.

Table 2-2 – Packaging Requirements

SRS #	Requirement	Description	Priority
4.1	Computer Hardware	The Smart Fitness trainer shall use a RGB-D camera to track the movement of the user.	1 - Critical
4.2	Software	The Smart Fitness Trainer shall analyze the user's movements to determine if the user is performing an exercise correctly.	1 - Critical
4.3	RGB-D Camera	The Smart Fitness Trainer shall allow the user to start Progression mode which will take the user through a medical check, fitness assessment, and allow the user to choose one of 3 fitness goals before generating a personalized workout.	1 - Critical
4.4	Heart-rate Device	The Smart Fitness Trainer shall perform a medical check which will have the user answer commonly asked questions by trainers before starting a new workout.	1 - Critical
4.5	Wireless Number Pad	The Smart Fitness Trainer shall have the user perform several basic exercises and measure the amount of correctly performed reps to provide a baseline of the user's prior workout experience.	1 - Critical
4.6	User Manual	The Smart Fitness Trainer shall allow the user to choose from one of three major workout goals: Strength, Tone, and Cardio.	3 - Moderate

2.2.3 Performance Requirements

This section provides a list of requirements detailing the performance of our system. The performance requirements will be used to create and implement performance standards such as point-cloud calculation times, system response times, and overall system efficiency.

Table 2-3 – Performance Requirements

SRS #	Requirement	Description	Priority
5.1	System Response Time	The Smart Fitness trainer shall make all calculations on data in a timely manner in order to provide real-time feedback.	1 - Critical
5.2	Video Processing Time	The Smart Fitness Trainershall process 640x480 resolution frames at 30 frames/second read in from the camera in a timely manner.	1 - Critical
5.3	Video Analysis/Tracking	The Smart Fitness Trainershall correctly analyze and track the user's movements in a timely-manner in order to provide sufficient system response time and video processing	1 - Critical

		time.	
5.4	Reliable Data Backup	The SMART Trainer system shall save and backup all data to the databases provided.	1 - Critical

2.2.4 Safety Requirements

This section describes the safety requirements of the Smart Fitness Trainer. The safety of the system and the person using the system are both covered in this section.

Table 2-4 – Safety Requirements

SRS #	Requirement	Description	Priority
6.1	Overheating Protection	The Smart Fitness Trainer shall include fans inside of the mini-computer's case to prevent the system from overheating.	1 - Critical
6.2	Non-Hazardous Setup	The Smart Fitness Trainer shall have minimal wire exposure. Only power cords and cords which connect the components (RGB-D Camera to computer) will be exposed.	1 - Critical
6.3	Health Warning	The Smart Fitness Trainer shall display a brief health warning as the program boots up	1 - Critical

2.2.5 Maintenance and Support Requirements

This section establishes the maintenance and support requirements for the Smart Fitness Trainer. These include the documentation of the code, as well as the testing and maintenance.

Table 2-5 – Maintenance and Support Requirements

SRS #	Requirement	Description	Priority
7.1	Code Documentation	The Smart Fitness Trainer source code shall be well-documented to allow future development teams to upgrade and add to the program.	1 - Critical
7.2	Testing	The Smart Fitness Trainer shall be tested to ensure all of the required functionality is present and working as intended.	1 - Critical
7.3	Troubleshooting Section	Team 4Loop shall include a troubleshooting section in the user manual to aid future developers and users with maintenance issues	1 - Critical
7.4	Product Extensibility	Team 4Loop shall set up the databases and code in a way that future development teams can easily add additional exercises and features to the system.	1 - Critical
7.5	Source Code	Team 4Loop shall provide the source code with the product to aid future developers, and allow the product to be added to easily.	1 - Critical

2.2.6 Other Requirements

This section will include requirements which do not fit in other categories, but are required for total completion of the Smart Fitness Trainer.

Table 2-6 – Other Requirements

SRS #	Requirement	Description	Priority
8.1	Achievements	The Smart Fitness Trainer shall award the user with achievements for completing certain objectives.	5 – Future
8.2	Workout Calendar	The Smart Fitness Trainer shall provide the user with a calendar view of their upcoming workouts.	3 - Moderate
8.3	Nutrition/Diet Plan	The Smart Fitness Trainer shall provide the user with sample diet/nutrition plans based on their workout goals.	5 – Future
8.4	Workout Drills Database	The Smart Fitness Trainer shall contain a database of workout drills that are categorized by muscle group, fitness goal, and degree of difficulty.	1 - Critical
8.5	Statistics Database	The Smart Fitness Trainer shall contain a database of total workout statistics for each user.	2 - High

2.3 Architecture Design Specification

The ADS document provided a high level design of the Smart Fitness Trainer by breaking the whole system down into five separate layers. Each layer was also broken into subsystems and data flows to further describe how the system is to be constructed. An overview of each architectural layer, as well as each subsystem and data flow it contains, will be given below. For more information, see the Architecture Design Specification.

2.3.1 Architecture Layer Overview

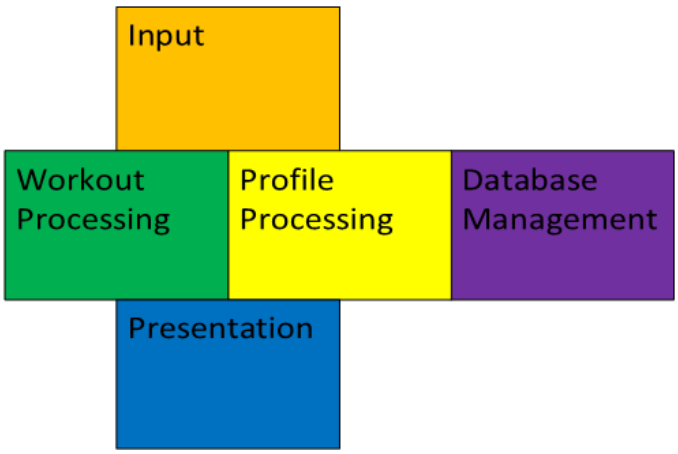

Figure 2-1 Architecture Layers

Input Layer

The Input Layer receives input from all of the hardware including the RGB-D camera, microphone, heart rate monitor, and wireless number pad. This layer will be responsible for converting and formatting these inputs to be used by the other layers. This will include reading the Bluetooth transmissions and parsing them to get meaningful values, and converting speech to text in order to process voice commands. This layer utilizes OpenNI to process the major points of a user's skeleton and create a hash map of these values.

Workout Processing Layer

The Workout Processing Layer focuses on initializing a workout and evaluating the user's performance on each drill in the workout. It will also need to keep track of user statistics including number of reps performed, correct reps completed, heart rate, and a timer to show how long the user has been working out.

Profile Processing Layer

The Profile Processing Layer is centered on things the user is doing while they are not currently working out. This layer allows a user to create, log in, and edit a profile, as well as view statistics from past workouts and create progression graphs. This layer also allows the user to view a workout's details and provides a calendar view of the workout. It must also tell the workout processing layer when the user wishes to start their workout.

Database Management Layer

The Database Management Layer is concerned with accessing the two databases (User Data and Drill Data). It will take requests from the Profile Processing Layer, convert them to queries, and return the requested data to the Profile Processing Layer. It will also be in charge of updating information in the database such as an edited profile or a user's new statistics after completing a workout.

Presentation Layer

The Presentation Layer receives data after it has been processed from either the Profile Processing or Workout Processing layers and displays the information to the screen. This layer is in charge of displaying the user interface while the user is navigating the menus and while the user is performing a workout. It will also need to send the audio to the user's speakers.

2.3.2 Subsystem Overview

Table 2-7 – Subsystem Overview

Layer Name	Subsystem Name	Description
Input	RGB-D Camera Input Handler	Collect input from RGB-D Camera
Input	Microphone Input Handler	Collect input from microphone
Input	Bluetooth Input Handler	Collect input from Bluetooth devices
Input	Voice-to-Text	Convent input from microphone to text
Workout Processing	Movement Comparison	Determine if user performs a rep of the current drill successfully.
Workout Processing	Workout Controller	Initialize the workout and process commands during the workout.
Workout Processing	Workout Instruction	Provide instructions for the current drill
Workout Processing	Statistic Tracker	Keep track of user statistics during a workout.
Profile Processing	Menu Navigation Controller	Process user commands while the user is not performing a workout.
Profile Processing	Post Workout Evaluation	Allow user to view statistics of previous workouts and create progression graphs.
Profile Processing	Profile	Allow user to create/edit/load a user profile.
Profile Processing	Workout Generator	Generate a workout based on the user's fitness level and fitness goal.
Database Management	User Data Access	Save/Load data to/from the user database
Database Management	Drill Data Access	Load data from the drill database
Presentation	Audio	Play audio on the speakers
Presentation	Menu Display	Display the user interface on the display

2.3.3 Inter-Subsystem Data Flow

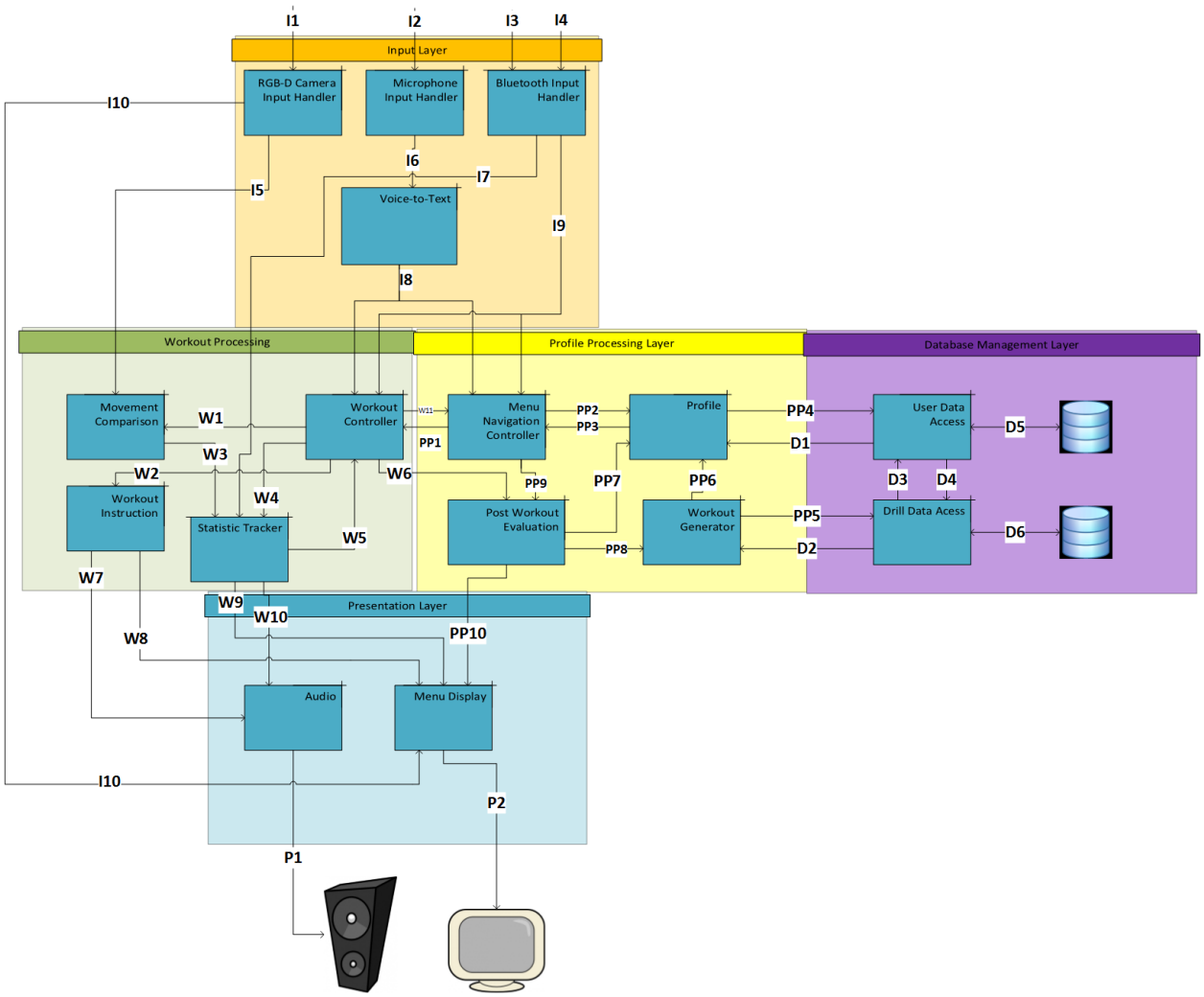

Figure 2-2 – Subsystem Data Flow Diagram

Table 2-8 – Subsystem Data Flows

Data Flow	Description
I1	Raw data from the RGB-D camera
I2	Raw audio data from microphone
I3	Raw unformatted Bluetooth transmission data from the heart rate monitor
I4	Raw unformatted Bluetooth transmission data from the wireless number pad
I5	Hash maps representing (x,y,z) positions of user's major skeletal joints
I6	Audio input to be converted to text in Voice-to-Text subsystem.
I7	Integer heart rate value to be sent to Statistic Tracker subsystem.
I8	Text Command to be sent to Workout Controller or Menu Navigation Controller (depending on if user is currently performing a workout)
I9	Number Pad Command to be sent to Workout Controller or Menu Navigation Controller (depending on if user is currently performing a workout)

I10	Depth Map from the RGB-D camera input handler to be sent to Camera Display subsystem
W1	Correct measurements for the current drill (hash maps with expected skeleton positions) sent to the movement comparison subsystem
W2	Audio and text instructions for the current drill loaded into the workout instruction subsystem
W3	Boolean value showing whether or not a rep was completed successfully
W4	Command to the statistic tracker to start the timer for a workout (used to track how long a user spends doing the drills)
W5	Updated rep counts/timer sent back to workout controller depending on what limit is set on the current drill (time based or rep based)
W6	Data structure containing stats from a completed or aborted workout for the current user
W7	Audio files are sent to the audio subsystem in the Presentation layer when the instructions are requested by the user.
W8	String containing text workout instructions are sent the menu display subsystem in the Presentation layer.
W9	Values of the tracked statistics are sent to the menu display subsystem in the Presentation layer.
W10	Request to play an audio feedback file is sent to the audio subsystem in the Presentation layer.
W11	Finish Workout function call to return to menu
PP1	Command to start performing a workout sent to the workout controller subsystem in the Workout Processing layer
PP2	Command to create or edit a profile sent to the profile subsystem
PP3	Data structure containing profile information such as name, statistics, and user's workout.
PP4	Requests to load or save profile information sent to the user data access subsystem in the Database Management layer
PP5	Requests to get drill information for drills which meet criteria for user based on fitness test results and workout goals.
PP6	Data structure representing a user workout
PP7	Data structure containing user statistics for a completed workout to be saved
PP8	Float value representing a user's score on the fitness test (0-10)
PP9	Command to display user's statistics.
PP10	User statistics in numeric or graph form depending on user selection.
D1	Requested structure containing user information
D2	List of drills matching requested criteria
D3	Drill information which matches the drill being pointed to
D4	Request for drill information that is being pointed to
D5	User information structure to be saved/loaded
D6	Drill information structure to be saved/loaded
P1	Audio output sent to speakers
P2	Video output of user interface to display

2.4 Detailed Design Specification

The DDS document took the information from the ADS document, and described it in much more detail by breaking down each subsystem into modules. These modules will be shown below, as well as a brief description of each data flow between the modules. For more information, see the Detailed Design Specification.

2.4.1 Module Overview

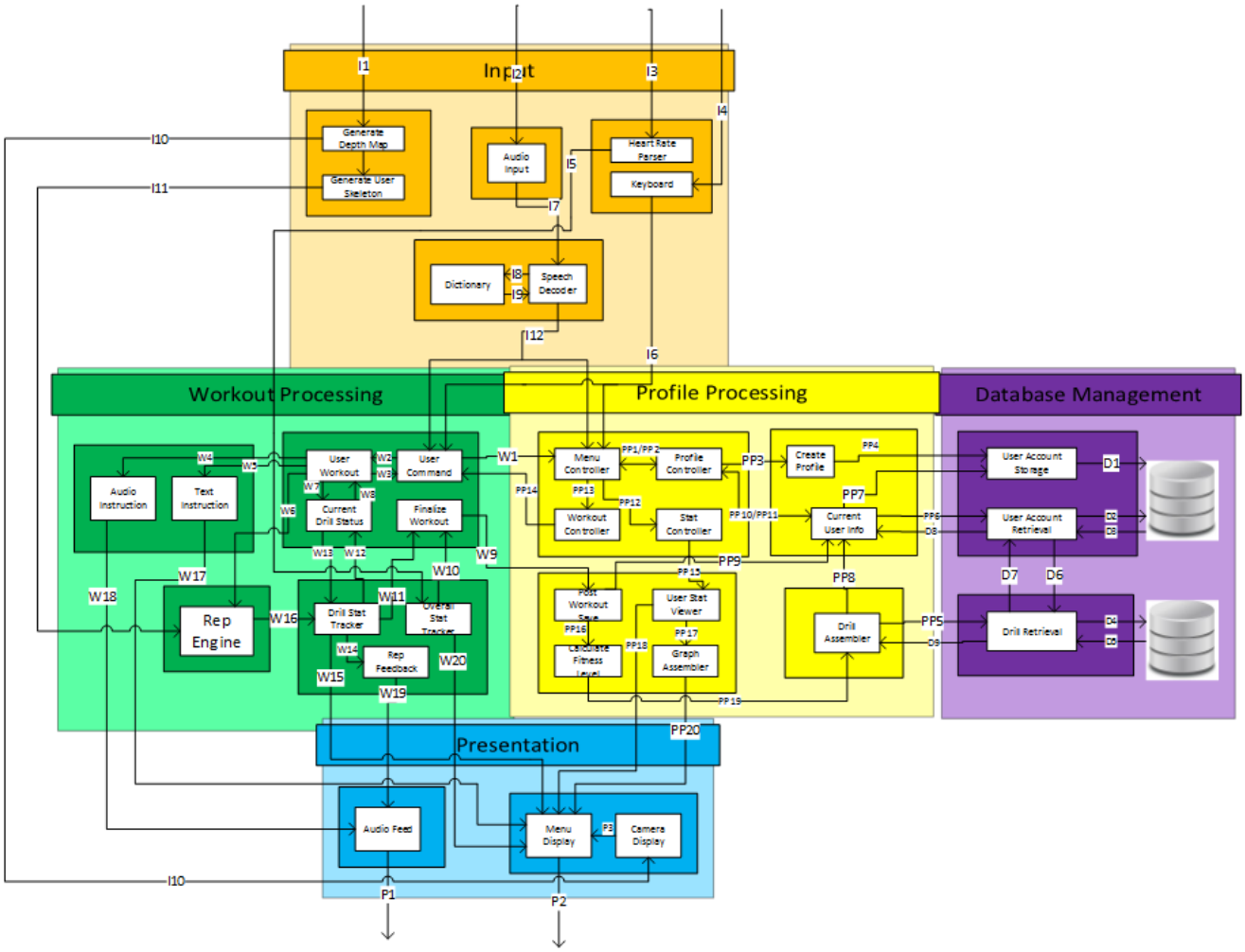

Figure 2-3 – Module Data Flow Diagram

2.4.2 Data Flow Descriptions

2.4.2.1 Input Layer Data Flows

Table 2-9 – Input Layer Data Flows

Data Flow	Description
I1	Raw video
I2	Raw audio.

I3	Bluetooth transmission containing heart-rate.
I4	Bluetooth transmission containing key press.
I5	Integer heart-rate value.
I6	Key press event containing the key that was pressed.
I7	Raw audio.
I8	Audio to attempt to match to a word.
I9	String matching the audio.
I10	Depth map.
I11	Hash map containing x,y,z coordinate of user's skeleton.
I12	Valid user command string.

2.4.2.2 Workout Processing Layer Data Flows

Table 2-10 – Workout Processing Layer Data Flows

Data Flow	Description
W1	Stop workout command.
W2	Command to call Start/Quit Workout function and a Workout object.
W3	Workout Complete Flag
W4	Audio file (.mp3) containing audio instructions of the current drill.
W5	String containing text instructions of the current drill.
W6	Hash map containing several correct positions of user skeleton for one rep of the current drill.
W7	Drill completion requirements. Amount of time or amount of reps needed for drill to be complete.
W8	Flag to tell User Workout to iterate to next drill in workout.
W9	Statistics structure from completed workout.
W10	Data structure containing overall stats for workout.
W11	Drill Statistic object for completed drill
W12	Drill ID and Integer representing current rep count or value of current drill timer.
W13	Command to reset drill rep counts or timer counts before moving to next drill in workout.
W14	Boolean rep result.
W15	Statistic object for current drill
W16	Boolean rep result.
W17	String to be displayed.
W18	The .mp3 file to be played.
W19	Audio file.
W20	Overall Statistics object

2.4.2.3 Profile Processing Layer Data Flows

Table 2-11 – Profile Processing Layer Data Flows

Data Flow	Description
PP1	Function call to instantiate menu.
PP2	Profile Data Structure
PP3	Function call to start profile creation.
PP4	New profile information to be saved.
PP5	Integer Fitness Level and String Fitness goal.
PP6	String Profile name.
PP7	User information to be saved.
PP8	User generated workout composed of drills.
PP9	Statistics structure from completed workout to be saved.
PP10	Function call to edit existing profile.
PP11	Profile Data Structure
PP12	Function call to instantiate menu.
PP13	Function call to instantiate menu.
PP14	Start Workout command and Workout object.
PP15	User statistics.
PP16	Fitness test statistics to be analyzed.
PP17	Request to create graph and user stats.
PP18	User Stats structure.
PP19	Integer fitness level.
PP20	Graph object to be displayed.

2.4.2.4 Database Management Layer Data Flows

Table 2-12 – Database Management Layer Data Flows

Data Flow	Description
D2	Request to load the user information
D3	User information structure to be loaded
D4	Request to load drill information
D5	Drill information structure to be loaded
D6	Request for drill information that is begin pointed to
D7	Drill information which matches the drill is being pointed to
D8	Data structure containing user information
D9	List of drills matching requested criteria

2.4.2.5 Presentation Layer Data Flows

Table 2-13 – Presentation Layer Data Flows

Data Flow	Description
P1	Audio file.
P2	Video source.
P3	JFrame containing Depth map

2.4.3 Module Requirements Mapping

Table 2-14 – Module Requirement Mapping

	Requirements	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	3.10	3.11	3.12	3.13	3.14	3.15	3.16
Input Layer	Heart Rate Parser			X		X											
	Keyboard			X	X	X	X			X			X				
	Audio Input			X	X	X	X			X			X			X	
	Dictionary			X	X	X	X			X			X			X	
	Speech Decoder			X	X	X	X			X			X			X	
	Generate Depth Map	X		X		X											
	Generate Skeleton	X		X		X											
Workout Processing Layer	User Command Controller			X		X				X							
	User Workout			X													
	Current Drill Status			X		X											
	Finalize Workout			X		X											
	Audio Instructions			X		X				X			X				
	Text Instructions			X		X				X			X				
	Overall Stat Tracker			X		X		X			X						
	Drill Stat Tracker			X		X		X			X						
	Rep Feedback			X		X											
	Rep Engine		X	X		X											

Profile Processing Layer	Menu Controller			X	X	X	X						X				
	Profile Menu Controller			X	X	X	X						X	X			
	Statistic Menu Controller												X				
	Workout Menu Controller			X		X						X	X				
	Create Profile			X	X	X	X							X			
	Current User Info			X										X	X		
	User Stat Viewer																X
	User Graph Assembler																
	Calculate Fitness Level			X		X			X								
	Post Workout Save			X		X					X						
	Drill Assembler			X			X		X								
DB Mgmt	User Account Storage			X							X						
	User Account Retrieval			X													
	Drill Retrieval			X					X								
Presentation	Menu Display			X	X	X	X			X			X				
	Camera Display			X		X											
	Audio Feed			X		X				X			X				

3. Test Items

3.1 Overview

This section will provide details on the different phases of testing which will occur before completing our final product. In the next sections, you will see our Relational Diagram which represents the System Test Plan as a flowchart. The STP is divided into five separate testing sections: hardware, unit, component, integration, and system verification.

Hardware testing will occur first to make sure the individual pieces of hardware that make up the system are functional. Next, we will put together tests for each module in unit testing. From unit testing, we can test an entire subsystem with component testing. All of the components in an entire layer will then be tested in integration testing. Finally, the entire system as a whole will be tested as a part of system verification testing.

3.2 Relational Diagram

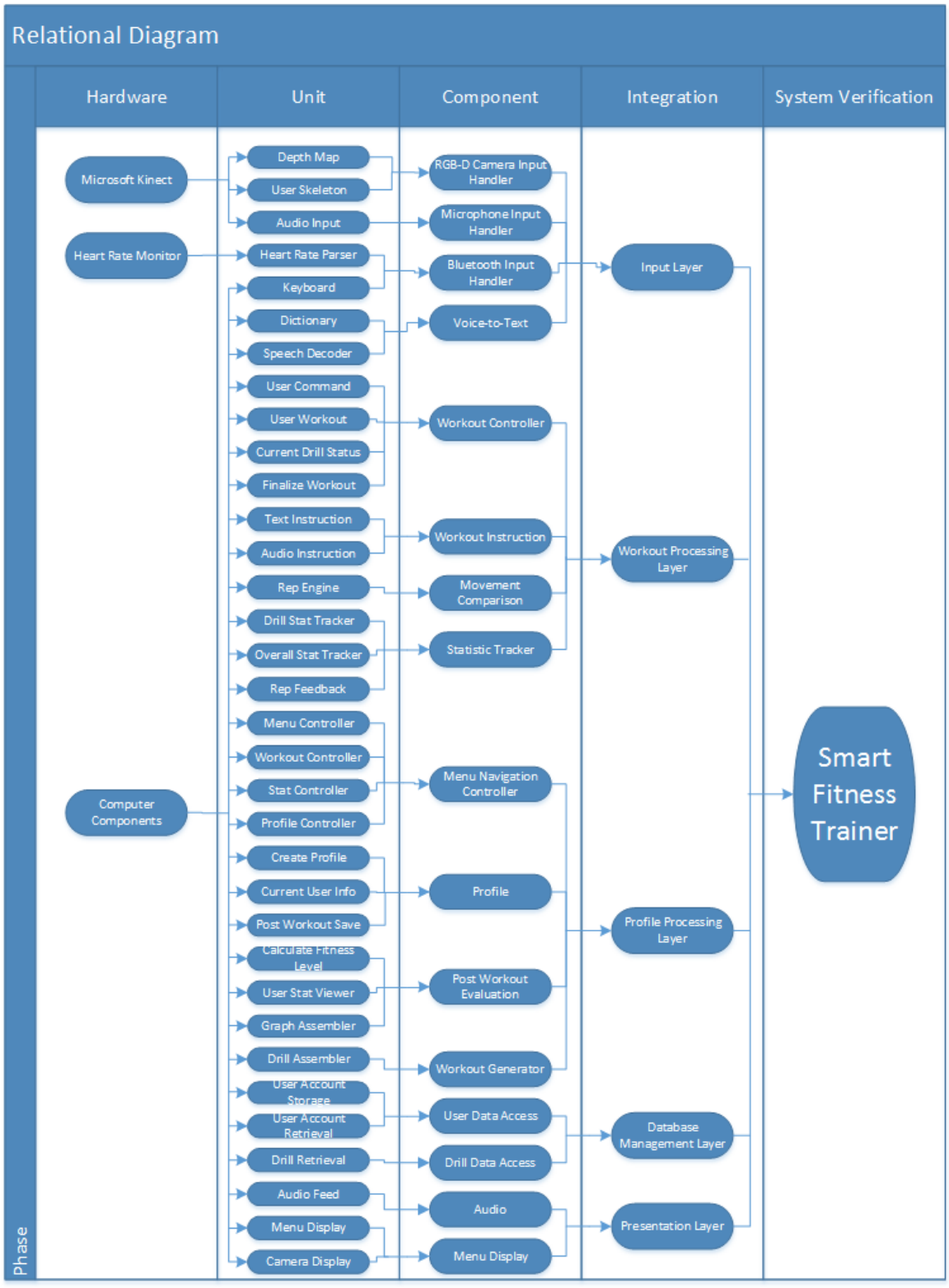

Figure 3-1 – Relational Diagram

3.3 Hardware Tests

Table 3-1 – Hardware Tests

Test ID	Module	Input	Expected Output/Action	Risk
H1	**Computer**	**User input via keyboard, microphone, RGB-D camera**	**Correct feedback to screen.** **Correct processing of Smart Fitness Trainer program**	**Medium**
H2	**Microsoft Kinect for Windows**	**Audio and video input**	**Delivers frames and audio input to the computer to be processed.**	**High**
H3	**Zephyr HxM Heart Rate Monitor**	**Bluetooth transmission**	**Bluetooth transmission sent to computer to be parsed for heart rate.**	**High**
H4	**Keyboard**	**Keyboard input**	**Correct key press sent to computer.**	**Low**

3.4 Unit Tests

3.4.1 Input Layer Unit Tests

Table 3-2 – Input Layer Unit Tests

Test ID	Module	Input	Expected Output/Action	Risk
UI1	**Generate Depth Map**	**Video Feed**	**Depth Map generated from camera feed.**	**High**
UI2	**Generate Skeleton**	**Depth Map**	**Skeleton Hash Map generated from Depth Map**	**High**
UI3	**Heart Rate Parser**	**Bluetooth transmission**	**Integer representing user's heart rate**	**High**
UI4	**Keyboard**	**Bluetooth transmission**	**Key press**	**Low**
UI5	**Audio Input**	**Audio source**	**Audio output from device**	**Medium**
UI6	**Dictionary**	**Audio output**	**Boolean flag and String value**	**High**
UI7	**Speech Decoder**	**Audio output from audio source**	**String matching the audio output if it was in the dictionary**	**High**

3.4.2 Workout Processing Layer Unit Tests

Table 3-3 – Workout Processing Layer Unit Tests

Test ID	Module	Input	Expected Output/Action	Risk
UWP1	**User Command**	**Action Event/String**	**Correct workout-related**	**Medium**

			function call depending on value of action event or string.	
UWP2	**User Workout**	**User Workout Object Start Workout Command Stop Workout Command**	**Iterate through the User Workout object (following all initialization steps for each drill) when the Start command is received, until the workout is completed, or the stop workout command is received.**	**Medium**
UWP3	**Current Drill Status**	**Drill Completion Requirement object Statistic Object**	**Compare incoming statistic object to the Drill Completion Requirement object to determine if drill is complete or not.**	**Medium**
UWP4	**Finalize Workout**	**OverallStat object Statistic object**	**Combine the OverallStat object and each Statistic object into the final OverallStat object.**	**Medium**
UWP5	**Text Instruction**	**String Text Instruction**	**Store current drill's text instruction file**	**Low**
UWP6	**Audio Instruction**	**Audio file containing audio instructions**	**Store current drill's audio instruction file**	**Low**
UWP7	**Rep Engine**	**User Skeleton Hash Map Correct Drill Instruction Hash Map**	**Compare values between the two hash maps and generate a Boolean value based on whether or not they match for each rep.**	**High**
UWP8	**Drill Stat Tracker**	**Boolean result Command to reset counter**	**Increment counters for current drill based on Boolean result obtained from rep engine. Reset counters to zero when command is received.**	**Medium**
UWP9	**Overall Stat Tracker**	**Integer heart rate value Timer object User information**	**Calculate calories burned based on timer object, heart rate, and user information.**	**Medium**
UPW10	**Rep Feedback**	**Boolean result**	**Determine to play feedback sound for good/bad rep Boolean value by using a random number generator.**	**Low**

3.4.3 Profile Processing Layer Unit Tests

Table 3-4 – Profile Processing Layer Unit Tests

Test ID	Module	Input	Expected Output/Action	Risk
UPP1	**Menu Controller**	**Action Event/String**	**Action Event or String should be sent to the correct menu**	**High**

			controller. Also, if Action Event or String is sent to Workout Controller it should also send a Workout object. If Action Event or String is sent to Statistics Controller it should also send an ArrayList of OverallStat objects.	
UPP2	**Profile Controller**	**Action Event String User Object**	**Send User object to Menu Controller Calls Create Profile Calls Current User Info**	**Medium**
UPP3	**Workout Controller**	**Action Event String Workout Object**	**Calls User Command module in Workout Processing layer with Workout object as parameters. Displays the details of Workout object**	**Medium**
UPP4	**Statistics Controller**	**Action Event String ArrayList of OverallStat objects**	**Calls User Stat Viewer with ArrayList of OverallStat objects are parameters**	**Medium**
UPP5	**Create Profile**	**Function call with no parameters**	**Conducts medical assessment, gives medical advice Calls User Account Storage module in Database Management layer with User object as parameter**	**Medium**
UPP6	**Current User Info**	**User Object Integer array of drill IDs OverallStat object**	**Calls User Account Storage module in Database Management layer with User object as parameter**	**High**
UPP7	**Post Workout Save**	**OverallStat object FitnessTestResults object Boolean variable**	**Calls Current User Info module with OverallStat object as parameter Calls Calculate Fitness Level module with FitnessTestResults object as parameter**	**Medium**
UPP8	**Calculate Fitness Level**	**FitnessTestResults object**	**Calls Drill Assembler module with integer as parameter**	**Medium**
UPP9	**User Stats Viewer**	**ArrayList of OverallStat objects**	**Calls Graph Assembler module with ArrayList of OverallStat objects as parameter Calls Menu Display module in Presentation layer with OverallStat object as parameter**	**Medium**
UPP10	**Graph Assembler**	**ArrayList of OverallStat objects**	**Calls Menu Display module in Presentation layer with integer array of data points as**	**Medium**

			parameter	
UPP11	**Drill Assembler**	**Integer representing fitness score** **Array of ten integers**	**Calls Drill Retrieval module in Database Management layer with fitness level (integer) and fitness goal (String) as parameters** **Calls Current User Info module with array of ten integers representing drill IDs as parameter**	**Medium**

3.4.4 Database Management Layer Unit Tests

Table 3-5 – Database Management Layer Unit Tests

Test ID	Module	Input	Expected Output/Action	Risk
UDB1	**User Account Storage**	**User object containing user information.**	**Create database queries to edit/create a user.**	**High**
UDB2	**User Account Retrieval**	**String containg the user's name.**	**Create database queries to load information about a user.**	**High**
UDB3	**Drill Retrieval**	**Integer representing fitness score and an array of integers.**	**Create database queries to retrieve a drill.**	**High**

3.4.5 Presentation Layer Unit Tests

Table 3-6 – Presentation Layer Unit Tests

Test ID	Module	Input	Expected Output/Action	Risk
UP1	**Menu Display**	**Integers for both the user statistics and drill statistic.s** **Strings that represent text instructions.**	**JFrame that will be comprised of the Menu Display's input.**	**High**
UP2	**Camera Display**	**Depth Map**	**Depth Map Jpanel displaying the video source.**	**High**
UP3	**Audio Feed**	**Audio File (.mp3/.wave)**	**Audio output to speaker.**	**Medium**

3.5 Component Tests

3.5.1 Input Layer Component Tests

Table 3-7 – Input Layer Component Tests

Test ID	Module	Input	Expected Output/Action	Risk
CI1	**RGB-D Camera Input Handler**	**Video Feed**	**Generate Depth Map to be displayed and User Skeleton Hash Map to be compared to Drill Specifications.**	**High**
CI2	**Microphone Input Handler**	**Audio source**	**Audio output from source**	**Low**
CI3	**Voice-to-Text**	**Audio output**	**String matching audio output if is found in the pre-defined dictionary.**	**High**
CI4	**Bluetooth Input Handler**	**Bluetooth transmissions**	**Integer heart rate and Key Press events**	**Medium**

3.5.2 Workout Processing Layer Component Tests

Table 3-8 – Workout Processing Layer Component Tests

Test ID	Subsystem	Input	Expected Output/Action	Risk
CWP1	**Workout Controller**	**ActionEvent/String to Start Workout**	**Makes the correct Profile subsystem function calls**	**Medium**
CWP2	**Workout Controller**	**Start Workout function call with Workout object parameter**	**Initializes the first drill of the workout by sending out audio/text instructions and correct drill movement hash maps**	**Medium**
CWP3	**Workout Controller**	**Statistic object OverallStat object**	**Combine OverallStat object and Statistic objects into single OverallStat object to be saved with user information**	**Medium**
CWP4	**Workout Controller**	**Integer containing current rep value or timer value**	**Compare to DrillCompletionRequirement for current drill to see if it is time to move to the next drill.**	**Low**
CWP5	**Workout Instruction**	**String Text Instructions Command to display instructions**	**Display current text instruction string when command to display is received.**	**Low**
CWP6	**Workout Instruction**	**Audio file with Audio Instructions Command to play instructions**	**Play current audio instruction file when command to play is received.**	**Low**
CWP7	**Movement Comparison**	**User Skeleton Hash Maps Correct Drill Movement Hash**	**Compare user skeleton hash maps to correct drill movement hash maps and produce Boolean value based on result.**	**High**

		Maps		
CWP8	**Statistic Tracker**	**Boolean value**	**Update current drill counters based on Boolean value received from Movement Comparison**	**Low**
CWP9	**Statistic Tracker**	**Integer Heart Rate User Information Timer Object**	**Update overall workout counters based on values from heart rate, user information, and timer**	**Medium**
CWP10	**Statistic Tracker**	**Function call to reset drill counters**	**Save final information for drill in a Statistic object and reset the current drill counters to zero.**	**Medium**

3.5.3 Profile Processing Layer Component Tests

Table 3-9 – Profile Processing Layer Component Tests

Test ID	Subsystem	Input	Expected Output/Action	Risk
CPP1	**Menu Navigation Controller**	**Action Event or String representing a selection from the profile menu**	**Makes the correct Profile subsystem function calls**	**Medium**
CPP2	**Menu Navigation Controller**	**Action Event or String representing a selection from the workout menu**	**Initializes a workout and outputs a user's workout. Displays the details of a user's workout**	**Medium**
CPP3	**Menu Navigation Controller**	**Action Event or String representing a selection from the statistic menu**	**Makes the correct Post Workout Evaluation subsystem function calls with ArrayList of OverallStat objects as parameter**	**Medium**
CPP4	**Profile**	**Function call with no parameters**	**Conduct Medical Assessment, receive user input values, function call to User Data Access subsystem with User object as parameter**	**Medium**
CPP5	**Profile**	**Function call with no parameter**	**Function call to User Data Access subsystem with user ID as parameter**	**Medium**
CPP6	**Profile**	**Function call with integer array as parameter**	**Function call to User Data Access with integer array as parameter**	**Medium**
CPP7	**Post Workout Evaluation**	**Function call with OverallStat object, FitnessTestResults object and Boolean variable as**	**If Boolean value is true it subsystem should make Profile function call with OverallStat object as the parameter. If Boolean value is false it should**	**Medium**

		parameters	**make Generate Workout function call with and integer representing a fitness-score as the parameter.**	
CPP8	**Post Workout Evaluation**	**Function call with ArrayList of OverallStat objects as parameter**	**If the function call specified a graph the subsystem should output an integer array. If it did not it should output an OverallStat object.**	**Medium**
CPP9	**Drill Assembler**	**Function call with integer as parameter**	**Should make call to the Drill Access subsystem with integer and String. It should receive an integer array and then make a call the Profile subsystem with integer array as parameter.**	**Medium**

3.5.4 Database Management Layer Component Tests

Table 3-10 Database Management Layer Component Tests

Test ID	Module	Input	Expected Output/Action	Risk
CDB1	**User Data Access**	**User object containg user information.**	**Updates and/or creates a user's information in the database.**	**High**
CDB2	**Drill Data Access**	**Array of integers.**	**Returns a specific drill.**	**High**

3.5.5 Presentation Layer Component Tests

Table 3-11 – Presentation Layer Component Tests

Test ID	Module	Input	Expected Output/Action	Risk
CP1	**Audio**	**Audio File (.mp3/.wave)**	**Receives the audio files that will be used to output to the user's speakers.**	**Medium**
CI2	**Menu**	**Integers for both the user statistics and drill statistics. Strings that represent text instructions. Depth Map that will contain the video source.**	**Create the UI that the user will use to navigate through the system. Output the JFrame to the user's display.**	**High**

3.6 Integration Tests

3.6.1 Input Layer Integration Tests

Table 3-12 – Input Layer Integration Tests

Test ID	Input	Expected Action/Result	Risk
II1	**Audio Source**	**Analysis of audio coming from source and string value that matches audio if is part of the pre-defined dictionary.**	**High**
II2	**Video Feed**	**Depth Map generation based on video and User Skeleton hash map generation based on depth map**	**High**

3.6.2 Workout Processing Layer Integration Tests

Table 3-13 – Workout Processing Layer Integration Tests

Test ID	Input	Expected Action/Result	Risk
IWP1	**Action event or string to start workout with a Workout object as a parameter.**	**Initialization of first drill from a workout object. (Load instruction files/drill specifications)**	**Medium**
IWP2	**Action event/string/flag to stop workout**	**Creation of OverallStat object for the workout**	**Medium**
IWP3	**Audio/Text Instruction Files Command to show instructions**	**String/Audio File output to display and speakers**	**Medium**
IWP4	**Boolean value representing rep result**	**Updated Statistic object**	**Low**
IWP5	**Integer heart rate New Timer object User information**	**Updated OverallStat object**	**Medium**
IWP6	**User Skeleton Hash Maps Drill Specification Hash Maps**	**Boolean value representing rep result**	**High**

3.6.3 Profile Processing Layer Integration Tests

Table 3-14 – Profile Processing Layer Integration Tests

Test ID	Input	Expected Action/Result	Risk
IPP1	**Action event or string to create profile**	**User object with new user's information**	**Medium**
IPP2	**Action event or string to login**	**A string of user's ID**	**Medium**
IPP3	**Action event or string to edit**	**User object with user's**	**Medium**

	profile	**new information**	
IPP4	**Action event or string to view statistics of workout**	**OverallStat object selected by user**	**Medium**
IPP5	**Action event or string to view progress of overall statistics**	**Integer array of coordinate points**	**Medium**
IPP6	**Action event or string to view workout**	**Workout object**	**Medium**
IPP7	**Action event or string to start workout**	**Workout object and function call to start workout**	**Medium**
IPP8	**OverallStat object as parameter**	**OverallStat object**	**Medium**
IPP9	**FitnessTestResults object**	**Integer fitness score and workout goal (string)**	**Medium**

3.6.4 Database Management Layer Integration Tests

Table 3-15 – Database Management Layer Integration Tests

Test ID	Input	Expected Action/Result	Risk
IDB1	**User object containing user information.**	**Retrieves and sends information back to the profile processing layer from the database.**	**Medium**
IDB2	**Integer fitness level and String fitness goal**	**Drills matching the fitness goal that are equal to or below the fitness level**	**Medium**
IDB3			

3.6.5 Presentation Layer Integration Tests

Table 3-16 – Presentation Layer Integration Tests

Test ID	Input	Expected Action/Result	Risk
IP1	**OverallStat object for the workout.**	**Update the Jlabel for the corresponding workout statistics.**	**High**
IP2	**Statistic object.**	**Update the Jlabel for the corresponding reps ad statistics for the user.**	**High**
IP3	**Audio instruction files.**	**Audio file output to the speakers.**	**Medium**
IP4	**Text instruction files.**	**Update the Jlabel for the corresponding instruction in a user's workout.**	**Medium**
IP5	**User object with information.**	**Update all JLabels for editing and/or creating a user.**	**High**

IP6	**String ID for username.**	**Update all Jlabels for the current user.**	**High**

3.7 System Verification Tests

Table 3-17 – System Verification Tests

Test ID	System Test	Input	Expected Output/Action	Risk
SV1	**Create a User Profile**	**Profile Information**	**Profile saved to database and selectable from main menu when system is turned on.**	**Low**
SV2	**Perform Fitness Test**	**Fitness Test Information, User Movements**	**User is given a score at the end of the fitness test based on how they perform the fitness test drills.**	**High**
SV3	**Generate Workout**	**Fitness Test Score, User Workout Goal**	**Workout is generated with up to ten drills based on fitness test score and workout goal.**	**Medium**
SV4	**Perform Workout**	**Drill Specifications, User Movements**	**User sees updated rep counts, timer, and workout statistics which accurately reflect how they perform movements of each drill.**	**High**
SV5	**View Statistics**	**Statistic objects, Heart Rate, Workout timer**	**User is able to view statistics from their previous workouts and generate progression graphs to view progress.**	**Medium**
SV6	**Voice Command Recognition**	**Audio giving valid voice command**	**System performs the specified function.**	**High**

4. Risks

4.1 Overview

This section identifies the risks involved during the testing phase, their impact, their severity and the strategy to handle them. Each risk will have a severity of high, medium or low.

4.2 Risk Table

Table 4-1 - Risks

ID	Risk	Impact	Severity	Strategy
1	Testing environment has faster processing speed than final product	Product passes performance test on testing hardware but runs slow on final product hardware	Medium	Make sure all hardware components in testing environment are of lesser or equal quality than final product
2	RGB-D camera loses track of user during testing	System will appear to fail to track movement	High	Make an error condition that notifies the user the camera has lost track of them the user can recalibrate the system
3	System responds positively to user movements when it should not	System will behave sporadically and it will be hard to tell if a test should pass or fail	High	While testing, print a message every time a user's movement causes the system to react to help pinpoint where any false response is coming from
4	Smaller modules pass tests but integrating them causes bugs	Bugs will be harder to find	Medium	Do regression testing each time a set of modules is integrated
5	Movement tracking might pass test for one user and not pass with another user	System will have inconsistent behavior	Medium	Everyone on the team should participate in the movement tracking testing
6	Heart rate monitor might not accurately be monitoring heart rate	User will not have accurate feedback from system	Medium	Attach two heart rate monitors to the user while testing and check to see they output the same values

5. Features to Be Tested

5.1 Overview

The following features will be tested to ensure that the Smart Fitness Trainer satisfies the requirements from the SRS. The descriptions provide the details of the requirement, while the test approach explains how we will verify its implementation.

5.2 Customer Requirements

5.2.1 RGB-D Camera Tracking

5.2.1.1 Risk

High

5.2.1.2 Description

The SMART Fitness trainer shall use an RGB-D camera to track the movement of the user.

5.2.1.3 Test Approach

We will test to make sure that the RGB-D Camera is generating a depth map by printing out the depth map to the screen.

5.2.2 User Movement Analysis

5.2.2.1 Risk

High

5.2.2.2 Description

The SMART Fitness Trainer shall analyze the user's movement to determine if the user is performing an exercise correctly.

5.2.2.3 Test Approach

We will test to make sure that the SMART Fitness Trainer is analyzing the user's movement correctly by displaying the user skeleton object to the screen and seeing if it matches the correct user position.

5.2.3 Workout Progression Mode

5.2.3.1 Risk

High

5.2.3.2 Description

The SMART Fitness trainer shall allow the user to start Progression mode which will take the user through a medical check, fitness assessment, and allow the user to choose one of 3 fitness goals before generating a personalized workout.

5.2.3.3 Test Approach

We will test this by verifying that all users are required to pass a medical check and fitness assessment during the creation of the user's Progression mode.

5.2.4 Medical Check

5.2.4.1 Risks

High

5.2.4.2 Description

The SMART Fitness trainer shall perform a medical check which will have the user answer commonly asked questions by trainers before starting a new workout.

5.2.4.3 Test Approach

We will test this by verifying that all users are required to pass a medical during the creation of the user's Progression mode.

5.2.5 Fitness Assessment

5.2.5.1 Risks

High

5.2.5.2 Description

The SMART Fitness trainer shall have the user perform several basic exercises, and measure the amount of correctly performed reps to provide a baseline of the user's prior workout experience.

5.2.5.3 Test Approach

We will test to make sure that the user is prompted for a fitness assessment before generating a workout. We will also verify that the correct drills are loaded for the user during a fitness assessment.

5.2.6 Workout Goals

5.2.6.1 Risks

Medium

5.2.6.2 Description

The SMART Fitness trainer shall allow the user to choose from one of three major workout goals: Strength, Tone, and Cardio.

5.2.6.3 Test Approach

We will test this by verifying that users have the option of selecting between three goals: Strength, Tone, and Cardio before their workout session.

5.2.7 Workout Statistics

5.2.7.1 Risks

High

5.2.7.2 Description

The SMART Fitness trainer shall track and display heart beat and calories burned (during the current workout) in real time.

5.2.7.3 Test Approach

We will test this by verifying that the correct heart-rate and estimated calories burned appear on the user's display. We will also test to make sure that the calories burned are of a reasonable number based on the equation used and the user’s information.

5.2.8 Personalized Workout

5.2.8.1 Risks

High

5.2.8.2 Description

The SMART Fitness trainer shall generate a personalized workout for the user based on the fitness assessment and workout goal.

5.2.8.3 Test Approach

We will test this by having a user perform a fitness assessment and have them select a workout goal. If a personalized workout has been generated, this requirement will have been tested.

5.2.9 Workout Instruction

5.2.9.1 Risks

Medium

5.2.9.2 Description

The SMART Fitness trainer shall provide both audio and visual instructions to the user on how to perform a drill, and what they should be doing at certain times during their workout (ex. rest period, start, stop, etc.)

5.2.9.3 Test Approach

We will test the audio by verifying that the correct audio audio file is being played during a drill along with the drill's text instructions appearing on the user's display.

5.2.10 Lifetime Workout Statistics

5.2.10.1 Risks

High

5.2.10.2 Description

The SMART Fitness trainer shall record the user's height, weight, BMI, total calories burned, minutes worked out, and other information about the user, which can be displayed in graph form.

5.2.10.3 Test Approach

We will test this by verifying that all statistics (height, weight, BMI, etc.) are correctly stored in the system database. We will test if the information can be displayed in graph form by giving sample coordinate data to ensure a graph is correctly constructed.

5.2.11 Quick Workout

5.2.11.1 Risks

Medium

5.2.11.2 Description

The SMART Fitness trainer shall allow the user to substitute any workout with a "quick workout", allowing them to select their own drills from the database for the workout.

5.2.11.3 Test Approach

We will test this by verifying that all users have the option to select 'Quick Workout' from the menu screen.

5.2.12 User Interface

5.2.12.1 Risks

High

5.2.12.2 Description

The SMART Fitness trainer shall have an appealing user interface which will be intuitive for the user. The interface shall include options such as selecting a profile, creating a profile, and selecting your workout.

5.2.12.3 Test Approach

We will test this by verifying that making sure that a minimum number of selectable options appear on each screen.

5.2.13 User Profile

5.2.13.1 Risks

High

5.2.13.2 Description

The SMART Fitness trainer shall allow the user to create, edit or delete a user profile, which will be used to "login" to progression mode.

5.2.13.3 Test Approach

We will test this by verifying that all users have the ability to selection those corresponding options (Create Profile, Edit Profile, etc.) on the system profile screen.

5.2.14 User Pictures

5.2.14.1 Risks

Low

5.2.14.2 Description

The SMART Fitness trainer shall allow the user to take pictures of themselves at different points in the workout plan to assess how the workout has impacted their appearance.

5.2.14.3 Test Approach

We will test this by having a test user take a sample picture.

5.2.15 Voice Commands

5.2.15.1 Risks

Low

5.2.15.2 Description

The Smart Fitness Trainer shall accept voice commands for menu interactions and other inputs during workouts.

5.2.15.3 Test Approach

We will test this verifying that the system responds to a user's voice command.

5.2.16 Personal Best Recognition

5.2.16.1 Risks

Medium

5.2.16.2 Description

The Smart Fitness Trainer shall be able to identify when a user has performed an exercise better than any previous attempts, and display a notification of a "personal best."

5.2.16.3 Test Approach

We will test this by having a test user with a pre-completed workout attempt undergo the same exercise routine to make determine whether the user has performed better on the exercise.

5.3 Performance Requirements

5.3.1 System Response Time

5.3.1.1 Risks

High

5.3.1.2 Description

The Smart Fitness Trainer shall calculate all point-cloud calculations and send all point-cloud transformation data in a timely manner in order to provide real-time feedback and statistics to the user.

5.3.1.3 Test Approach

We will test this verifying that all commands and functions can be performed in an appropriate amount of time in test run-through without any lag and/or system freezes.

5.3.2 Video Processing Time

5.3.2.1 Risks

High

5.3.2.2 Description

The Smart Fitness Trainer shall process 640x480 resolution frames at 30 frames/second read in from the camera in a timely manner.

5.3.2.3 Test Approach

We will test this by verifying that the system can output the video source grabbed from the RGB-D camera to the user's display without any noticeable lag and/or freezes.

5.3.3 Video Analysis/Tracking

5.3.3.1 Risks

High

5.3.3.2 Description

The Smart Fitness Trainer shall correctly analyze and track the user's movements in a timely-manner in order to provide sufficient system response time and video processing time.

5.3.3.3 Test Approach

We will test this by verifying that hash maps are formed during video analysis/tracking. These hash maps will be compared to drill specification hash maps and update rep counts in an adequate amount of time.

5.3.4 Background Noise Reduction

5.3.4.1 Risks

Low

5.3.4.2 Description

The Smart Fitness Trainer shall differentiate between background noise and voice commands given from the current user.

5.3.4.3 Test Approach

We will test this verifying that the system can interpret voice commands and setting with various sounds.

5.3.5 Reliable Data Backup

5.3.5.1 Risks

Low

5.3.5.2 Description

The SMART Trainer system shall save and backup all data that have been both received and generated for the user throughout his/her workout regimen in order to provide both a progression and statistical-based approach for the user.

5.3.5.3 Test Approach

We will test this by verifying that all saved data can be retrieved.

5.4 Maintenance and Support Requirements

5.4.1 Testing

5.4.1.1 Risks

5.4.1.2 Description

The Smart Fitness Trainer shall be tested to ensure all of the required functionality is present and working as intended.

5.4.1.3 Test Approach

We will test this by verifying that all required functionality can be provided.

5.5 Other Requirements

5.5.1 Achievements

5.5.1.1 Risks

Medium

5.5.1.2 Description

The Smart Fitness Trainer shall award the user with achievements for completing certain objectives. These achievements will include milestones such as X number of total calories burned, completing a four week workout program, and several others.

5.5.1.3 Test Approach

We will test this by having a user objective in place for a test user to achieve. If the test user achieves this objective, this requirement would have been fulfilled.

6. Features Not to Be Tested

6.1 Overview

The following features listed are not to be tested since they are verified by system design. These features describe the system properties of the product and do not much functionality.

6.2 Packaging Requirements

All listed items will be visually verified by project members to verify all items are packaged with the system.

6.2.1 Computer Hardware

6.2.1.1 Description

Computer hardware shall be delivered in a mini-computer. The mini-computer will include the basic hardware components needed to run the Smart Fitness Trainer software: hard-drive, RAM, graphics card, processor, fans, power supply and motherboard. The hardware will come pre-assembled and enclosed in a micro-case.

6.2.2 Software

6.2.2.1 Description

The Smart Fitness Trainer software shall be pre-installed on the mini-computer's hard-drive.

6.2.3 RGB-D Camera

6.2.3.1 Description

System shall include an RGB-D camera. Camera will come with a cord to connect with the mini-computer.

6.2.4 Heart-Rate Device

6.2.4.1 Description

System shall include a device to keep track of user's heart-rate during a work-out.

6.2.5 Wireless Number Pad

6.2.5.1 Description

System shall include a wireless number pad for users to navigate menus and make selections.

6.2.6 User Manual

6.2.6.1 Description

System shall include a user manual with step-by-step instructions on how to setup and use the Smart Fitness Trainer. User manual will include pictures to aid user.

6.3 Safety Requirements

The safety of items are physical properties that are built to verify that the overall system contains no harm to the user and/or to prevent any tampering.

6.3.1 Overheating Protection

6.3.1.1 Description

The Smart Fitness Trainer shall include fans inside of the mini-computer's case to prevent the system from overheating.

6.3.2 Non-Hazardous Setup

6.3.2.1 Description

The Smart Fitness Trainer shall have minimal wire exposure. Only power cords and cords which connect the components will be exposed.

6.3.3 Health Warning

6.3.3.1 Description

The Smart Fitness Trainer shall display a brief health warning as the program boots up. This warning's purpose will be to recommend all participants see a doctor before starting a new exercise program.

6.4 Other Requirements

All other requirements listed here are not to be tested since they are verified by system design and implementation.

6.4.1 Workout Calendar

6.4.1.1 Description

The Smart Fitness Trainer shall provide the user with a calendar view of their upcoming workouts. Each day shall have an icon corresponding to the type of workout for that day. The calendar will show all of the workouts for a user over the four week workout plan provided to

6.4.2 Nutrition/Diet Plan

6.4.2.1 Description

The Smart Fitness Trainer shall provide the user with sample diet/nutrition plans based on their

workout goals.

6.4.3 Workout Drills Database

6.4.3.1 Description

The Smart Fitness Trainer shall contain a database of workout drills that are categorized by muscle group, fitness goal, and degree of difficulty.

6.4.4 Statistics Database

6.4.4.1 Description

The Smart Fitness Trainer shall contain a database of total workout statistics for each user.

7. Overall Test Strategy

7.1 Overview

This section shall detail team 4Loop's approach to testing the Smart Fitness Trainer prototype. The testing should ensure that the prototype meets the requirements set forth in the System Requirements Specification document and is consistent with the design in the Architectural Design Specification document and the Detailed Designed Specification.

7.2 Configurations

Team 4Loop will test configurations in the order of high priority to low priority. This means that no configurations of a lower priority will take place until all higher priority configurations have passed testing.

7.3 Strategy

Testing will be performed in a series of stages. The system will be broken down into its smallest units and then each unit will be tested individually. When each unit has passed testing the units will be integrated and tested as a whole. The following list shows each stage of testing in order:

- Hardware Testing
- Unit Testing
- Component Testing
- Integration Testing
- System Verification Testing

7.4 Metrics

The priority of each test will be determined by team 4Loop by associating each test with a requirement in the System Requirements Specification document. The tests will have the same priority level as the requirement it is associated with. The priority levels are high, medium and low.

7.4.1 – High

A failure of a test with a priority level of high means that at least one high priority requirement in the System Requirements Specification will not be met. Because these requirements are critical to the acceptance of the system a redesign of the system or replacement of the component may be necessary.

7.4.2 – Medium

A failure of a test with a priority level of medium means that at least one medium priority requirement in the System Requirements Specification will not be met. Because these requirements are important but not critical to the acceptance of the system a redesign of the system or replacement of the component will probably not be necessary.

7.4.2 – Low

A failure of a test with a priority level of low means that at least one low priority requirement in the System Requirements Specification will not be met. Because these requirements are not very important and not critical to the acceptance of the system a redesign of the system or replacement of the component will not be considered.

7.5 Regression

Regression testing will be conducted after a stage of testing has been completed and smaller modules are integrated together. For instance, regression testing will be done when unit testing is completed and the units are integrated together to form components. This will ensure that no new bugs are introduced as a result of integration.

8. Acceptance Criteria

8.1 Overview

The Acceptance Criteria will be used to determine if a test for a particular item is a Pass or a Fail. These criteria for each Pass or Fail option has also been divided into five categories: Hardware, Unit, Component, Integration, and System Verification.

8.2 Hardware Testing

All of the computer hardware will be tested when the pieces have been assembled into the case. All other hardware will be tested individually

Pass

The hardware unit returns the expected output when given a valid input. The hardware unit returns an error when given an invalid input.

Fail

The hardware unit fails to provide the expected output when given a valid input. The hardware unit fails to provide an error when given an invalid input.

8.3 Unit (Module) Testing

The given inputs and expected outputs are based on the documentation in the DDS document.

Pass

The module returns the expected output when given a valid input or returns an error when given an invalid input.

Fail

The module fails to provide the expected output when given a valid input or fails to provide an error when given an invalid input.

8.4 Component (Subsystem) Testing

When all modules in a subsystem have passed Unit Testing, that subsystem as a whole will be tested. The given inputs and expected outputs are based on the documentation in the ADS and DDS documents.

Pass

The subsystem returns the expected output when given a valid input or returns an error when given an invalid input.

Fail

The subsystem fails to provide the expected output when given a valid input or fails to provide an error when given an invalid input.

8.5 Integration (Layer) Testing

After all subsystem in a layer have passed Component Testing, that layer as a whole will be tested. The given inputs and the expected outputs for Integration Testing are based off documentation in the ADS and DDS documents.

Pass

The layer returns the expected output when given a valid input or returns an error when given an invalid input.

Fail

The layer fails to provide the expected output when given a valid input or fails to provide an error when given an invalid input.

8.6 System Verification Testing

After each layer in the system has passed Integration Testing, the entire Smart Fitness Trainer system will be tested. The given inputs and expected outputs for System Verification Testing are based off documentation in the SRS, ADS, and DDS documents.

Pass

The Smart Fitness Trainer system returns the expected output when given a valid input, and meets all related requirements in the SRS document. The system returns an error when given an invalid input.

Fail

The Smart Fitness Trainer system fails to provide the expected output when given a valid input or does not meet all related requirements in the SRS document. The system fails to provide an error when given an invalid input.

9. Test Deliverables

9.1 Overview

This section will discuss all of the deliverables included with the end project documentation which pertain to the system testing.

9.2 Deliverables

9.2.1 System Test Plan

This document will provide the user with the necessary information to fully describe the process used by our team to perform the testing of the product.

9.2.2 Test Cases

This deliverable will include a description of each test case used by our team. Information provided for each test case will also include:

- Test Case ID
- Test Date
- Tester Name
- Inputs
- Expected Output
- Actual Output
- Result of Test (Pass/Fail)
- Additional Notes

If a defect is found in the test case, the defect ID and severity will be included in the additional notes section.

9.2.3 Defects

If a test has a 'Fail' test result, there will be a defect description provided. The information for each defect will include:

- Test Case ID
- Test Date
- Tester Name
- Description of Failure
- Status
- Severity (High/Medium/Low)
- Effects on development
- Additional Notes

9.2.4 Test Code

Any additional code used for testing the final product will also be provided with the end project documentation.

10. Test Schedule

10.1 Overview

This section contains the schedule for the testing process of the Smart Fitness Trainer system. This schedule is also reflected in our MS Project Plan document.

10.2 Test Schedule

Table 10-1 – Test Schedule

Task Number (Project Plan)	Task	Planned Start Date	Planned Finish Date
2.5	Hardware Testing	07/11/13	07/14/13
2.6	Unit Testing	07/15/13	07/20/13
2.7	Component Testing	07/21/13	07/26/13
2.8	Integration Testing	07/27/13	08/02/13
2.9	System Verification Testing	08/03/13	08/08/13

11. Approvals

This section provides the approval of this document's presented plan by all necessary individuals who are part of the project.

Table 11-1 - Approvals

Name	Title	Signature	Date
Mike O'Dell	Project Supervisor		
Jeremy Roden	Project Sponsor		
Thomas Heinen	Team Lead		
Thanuja Fernando	Team Member		
Andrew Gallagher	Team Member		
Mark Ragunton	Team Member		